

1. Введение в Godot

Что такое Godot?

Godot — это универсальный 2D и 3D игровой движок, спроектированный для поддержки всех видов проектов. Вы можете использовать его для создания игр или приложений, которые можете затем выпускать на настольных или мобильных платформах, а также web.

Вы также можете создавать игры для приставок, хотя это требует высокого навыка программирования или разработчика, который портирует игру за вас.

Что может движок?

Godot изначально был разработан аргентинской игровой студией. Его разработка началась в 2001 году, а движок был переписан и значительно улучшен с момента его выпуска с открытым исходным кодом в 2014 году.

К примерам игр, созданных в Godot, относятся Ex-Zodiac и Helms of Fury.

Что касается приложений, то программа для рисования пиксельной графики с открытым исходным кодом - Pixelorama работает на базе Godot, как и RPG in a box.

Вы можете найти намного больше примеров в [официальных демонстрационных видео](#).

Обзор ключевых концепций Godot

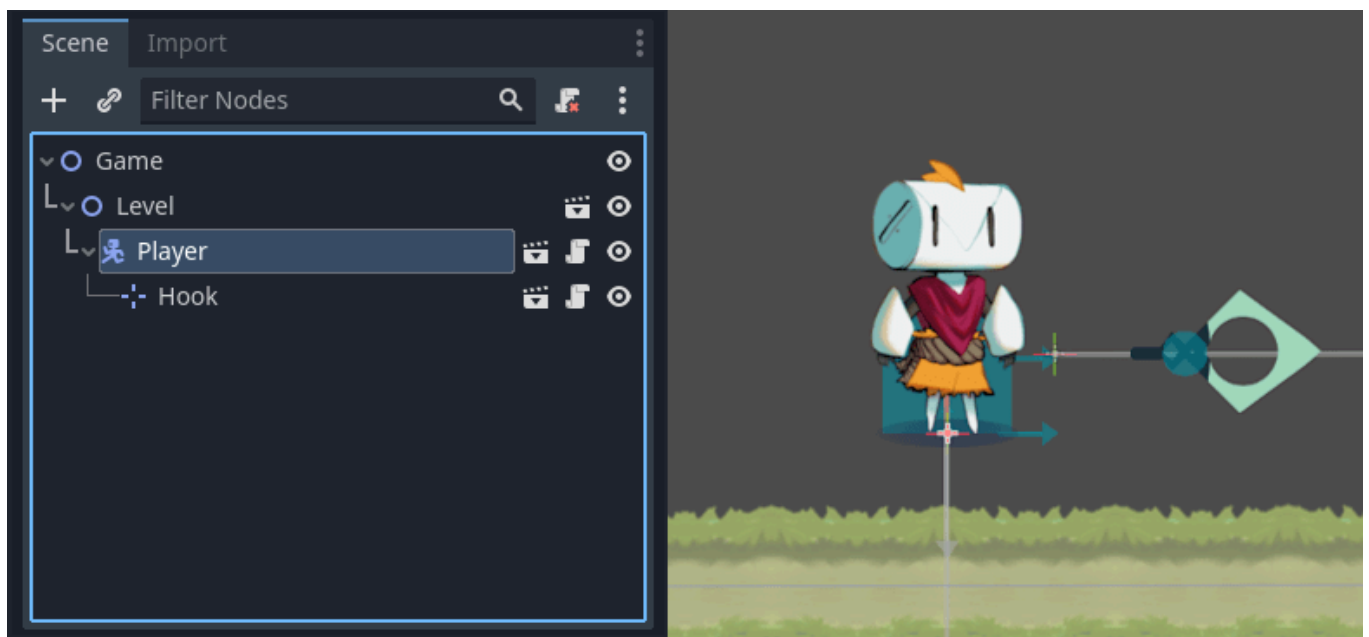
Каждый игровой движок основывается на абстракциях, которые используют для сборки приложений. В Godot игра - это **дерево узлов**, сгруппированных в **сцены**, которые могут обмениваться информацией с помощью **сигналов**.

Есть четыре основные концепции, с которыми вы познакомитесь. Мы рассмотрим их кратко, чтобы дать представление того, как работает движок. В цикле «Начало работы» вы воспользуетесь ими на практике.

Сцены

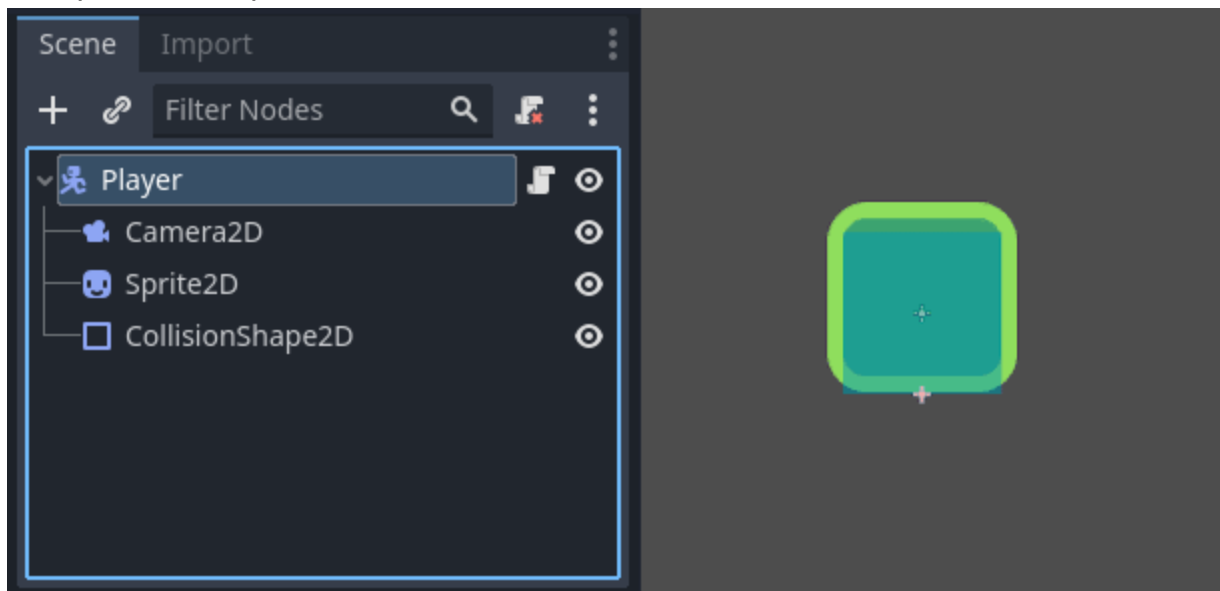
В Godot вы разбиваете свою игру на многократно используемые сцены. Сцена может быть персонажем, оружием, меню пользовательского интерфейса, отдельным домом, целым уровнем или всем, что вы можете придумать. Сцены Godot гибки; они одновременно выполняют роль префабов и сцен из некоторых других игровых движков.

Вы также можете вкладывать сцены. Например, вы можете поместить своего персонажа на уровень и перетащить сцену как его дочерний элемент.



Узлы

Сцена состоит из одного или большего числа **узлов**. Узлы — самые маленькие строительные блоки вашей игры, которые вы выстраиваете в деревья. Это пример узла, содержащего персонажа.



Здесь он состоит из узла `CharacterBody2D` с именем "Player", `Camera2D`, `Sprite2D` и `CollisionShape2D`.

Имена узлов заканчиваются на "2D" из-за того, что это 2D-сцена. Имена их аналогов в 3D заканчиваются на "3D".

Заметьте, как одинаково выглядят узлы и сцены в редакторе. Когда вы сохраняете дерево узлов как сцену, она отображается одиночным узлом, внутренняя структура которого скрыта в редакторе.

Godot предоставляет обширную библиотеку базовых типов узлов, которые вы можете совмещать и расширять для создания более функциональных. 2D, 3D или пользовательский интерфейс — вы будете делать большинство вещей при помощи этих узлов.

Дерево сцены

Все сцены вашей игры собираются вместе в **дереве сцены**, буквально дереве из сцен. И, так как сцены — деревья узлов, дерево сцены также является деревом узлов. Но проще воспринимать вашу игру с точки зрения сцен, поскольку они могут представлять персонажей, оружие, двери или пользовательский интерфейс.

Сигналы

Узлы транслируют сигнал, когда происходит какое-то событие. Эта возможность позволяет задать взаимодействие узлов без жёсткого указания их в коде. Это даёт вам большую гибкость в структурировании сцен.

Например, кнопка испускает сигнал при нажатии. Вы можете подключить такой сигнал для выполнения кода в ответ на это событие, к примеру, для начала игры или открытия меню.

Другие встроенные сигналы могут сообщить вам о столкновении двух объектов, о том, что персонаж или монстр вошёл в заданную область, и многом другом. Вы также можете определить новые сигналы, адаптированные к вашей игре.

Подведение итогов

Узлы, сцены, дерево сцен и сигналы — четыре ключевых понятия а Godot, которыми вы будете оперировать постоянно.

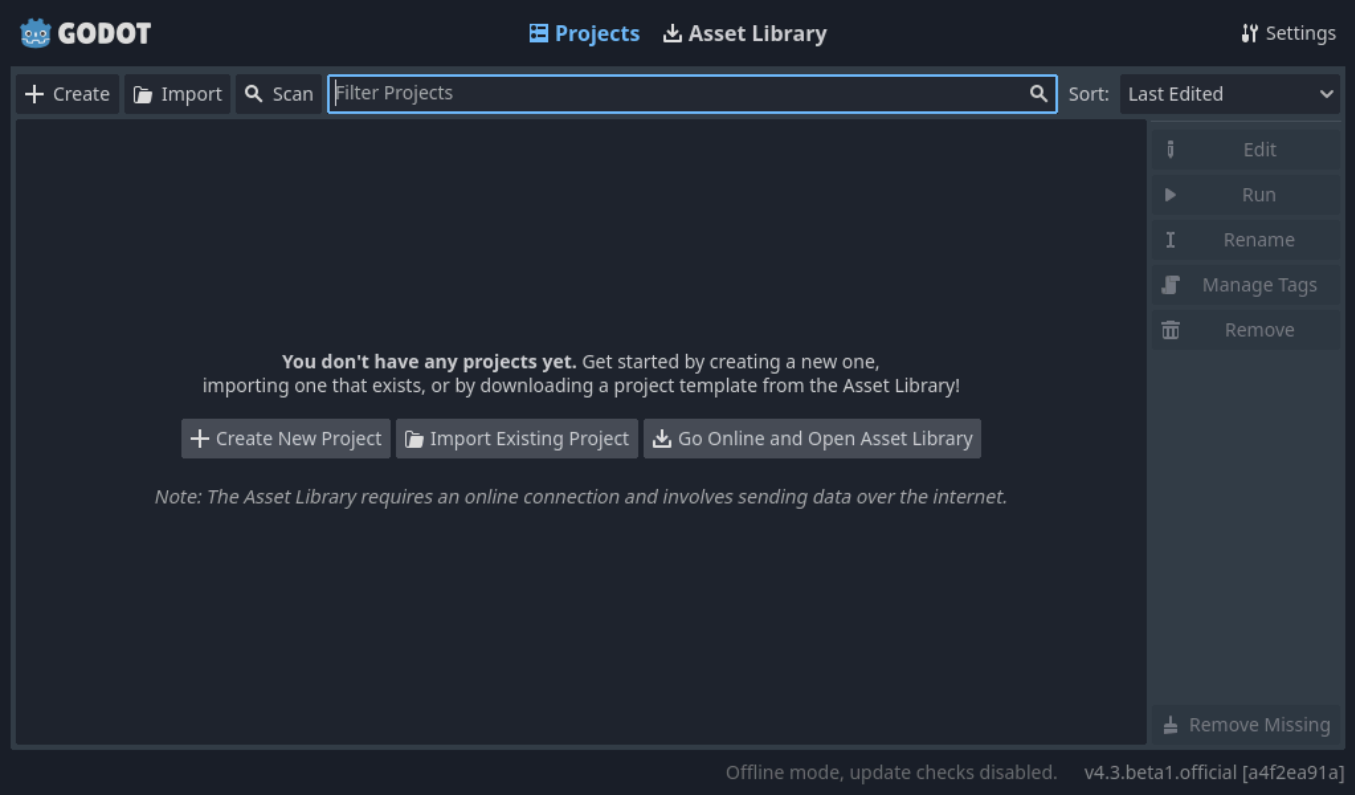
Узлы — это самые маленькие строительные блоки вашей игры. Вы комбинируете их для создания сцен, которые, в свою очередь, объединяются и вкладываются в дерево сцен. Вы можете использовать сигналы для задания реакции узла на события в других узлах или различных ветвях дерева сцены.

После этого краткого обзора, вы, наверное, имеете множество вопросов. Потерпите, так как вы получите ответы на многие вопросы на протяжении цикла «Начало работы».

Первый взгляд на редактор Godot

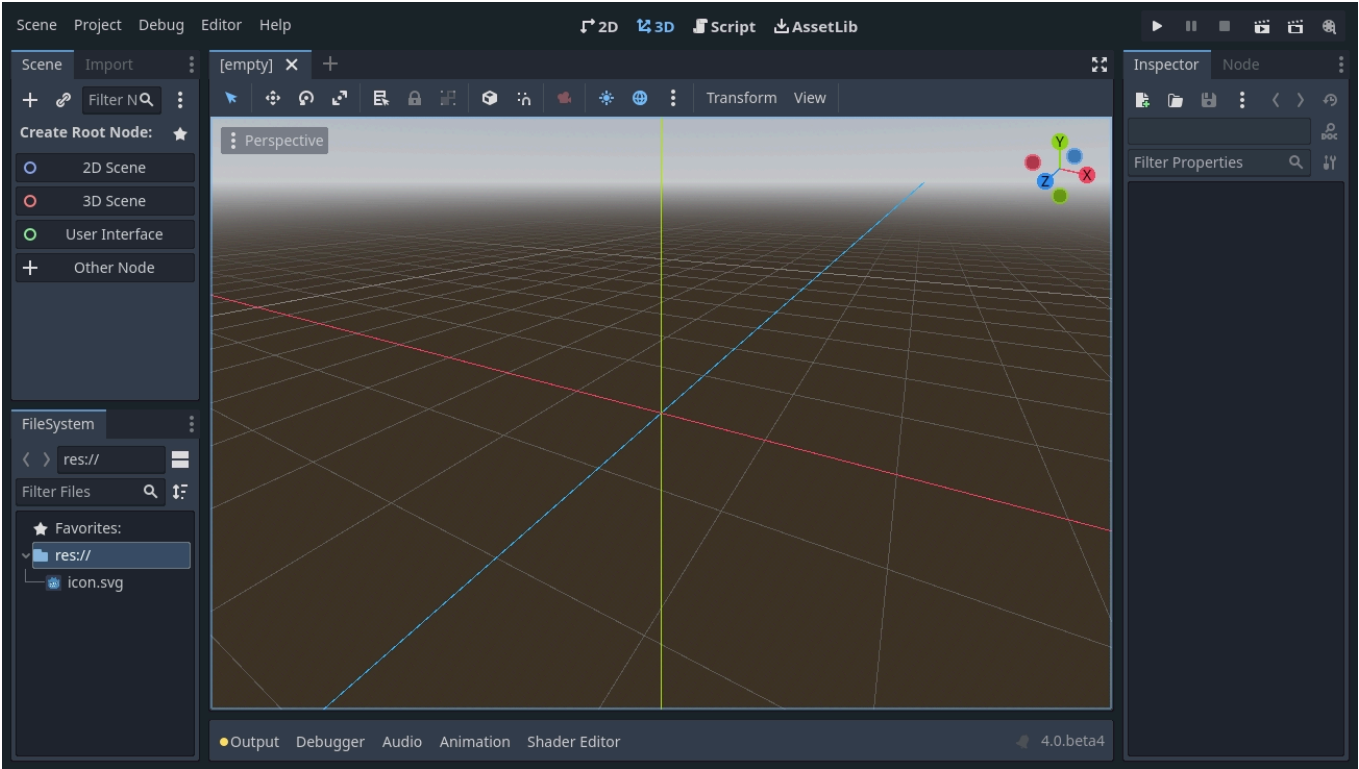
Менеджер проектов

Когда вы запускаете Godot, первое окно, которое вы видите - это Менеджер проектов. На вкладке по умолчанию "Локальные проекты" вы можете управлять существующими проектами, импортировать или создавать новые и многое другое.



Первый взгляд на редактор Godot

Когда вы открываете новый или существующий проект, появляется интерфейс редактора. Давайте взглянем на основные области.



По умолчанию он имеет **меню, основные экраны** и кнопку тестового запуска на верхнем крае окна.

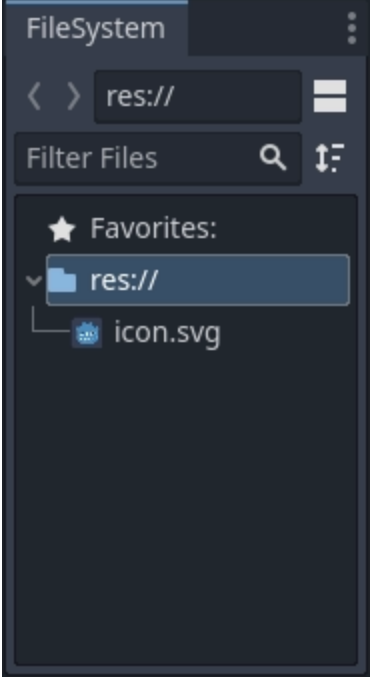


В центре находится **окно просмотра** с его **панелью инструментов** сверху, где вы сможете найти инструменты для перемещения, масштабирования или блокировки узлов сцены. Панель изменяется, когда вы переключаетесь между различными рабочими областями.

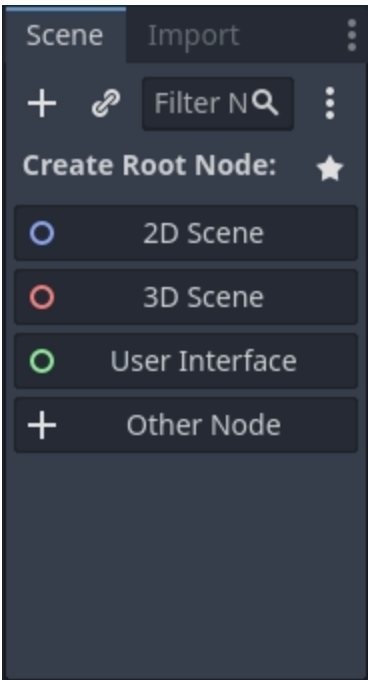
Изменения панели инструментов зависит от контекста и выбранного узла. Это панель инструментов для 2D.



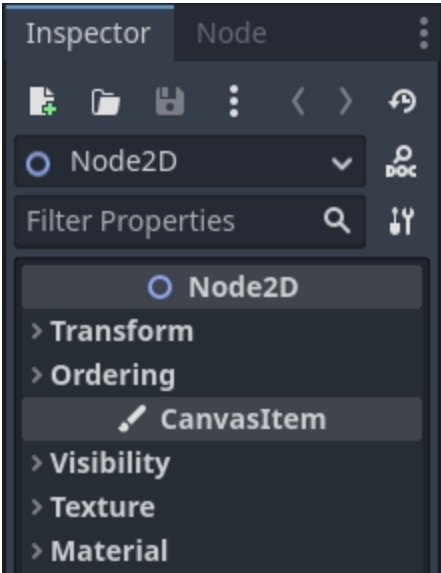
Давайте взглянем на панели. В панели «Файловая система» перечислены файлы вашего проекта, включая скрипты, изображения, аудио и многое другое.



Панель **Scene (Сцена)** отображает активные узлы сцены.



Inspector (Инспектор) позволяет вам редактировать свойства выбранного узла.



Нижняя панель под окном просмотра включает в себя консоль отладки, редактор анимации, аудиомикшер и многое другое. Они широкие и могут занимать много драгоценного пространства, поэтому по умолчанию они свернуты.



Когда вы нажимаете на один из них, он вертикально расширяется. Ниже вы можете увидеть открытый редактор анимации.

Четыре основных экрана

В верхней части редактора расположены четыре кнопки рабочего пространства: 2D, 3D, Script и AssetLib.

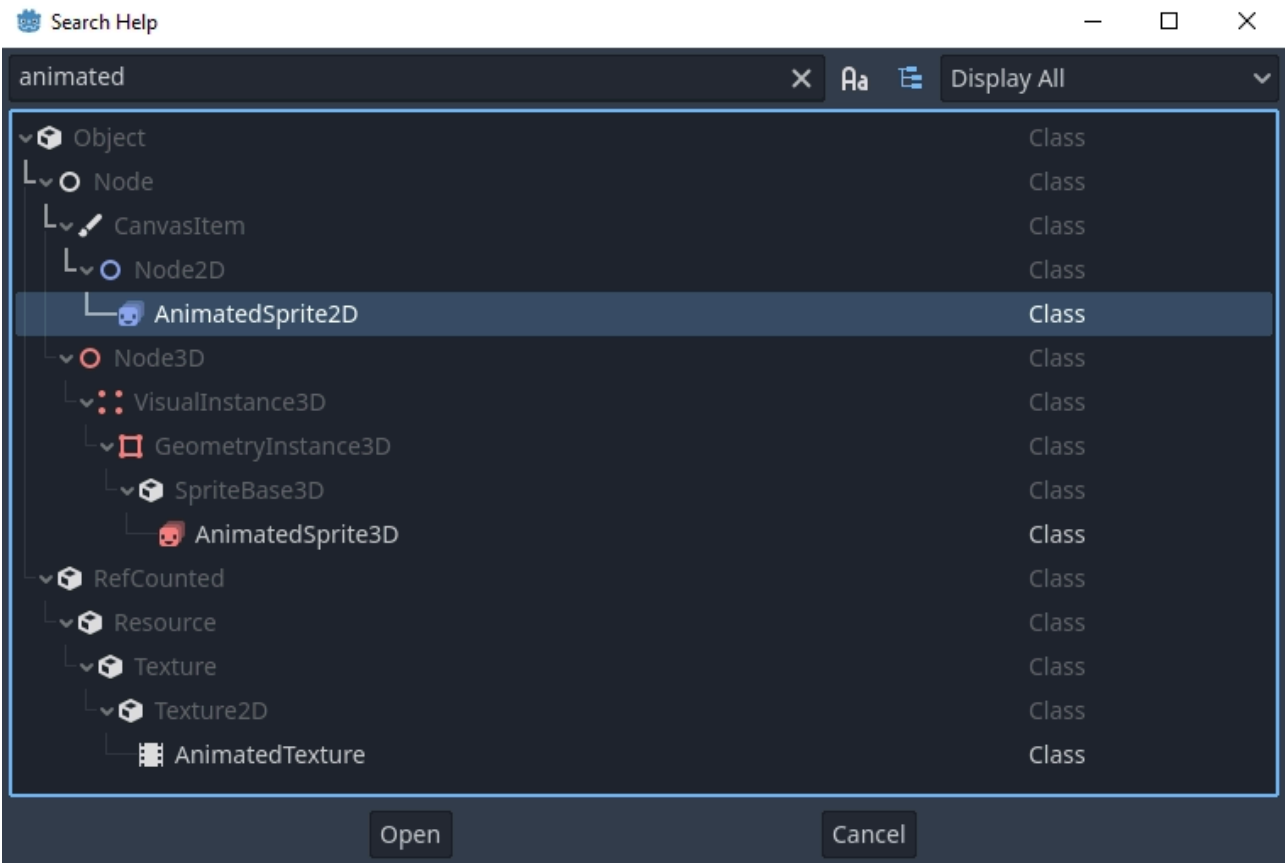
- Вы будете использовать **2D-пространство** для всех типов игр. Кроме непосредственно 2D-игр, там вы будете создавать свои интерфейсы. Нажмите Ctrl + F1 (или Alt + 1 на macOS) для доступа к нему.
- В **3D-пространстве** вы можете работать с сетками (meshes), освещением и дизайном уровней для 3D-игр. Нажмите Ctrl + F2 (или Alt + 2 на macOS) для доступа к нему.
- Экран **Script** — это полноценный редактор кода с отладчиком, богатым автозаполнением и встроенной справкой по коду.
- Наконец, **AssetLib** - это библиотека бесплатных дополнений, скриптов и ассетов для дальнейшего использования в ваших проектах.

Справочник по интегрированным классам

Godot поставляется со встроенным справочником классов.

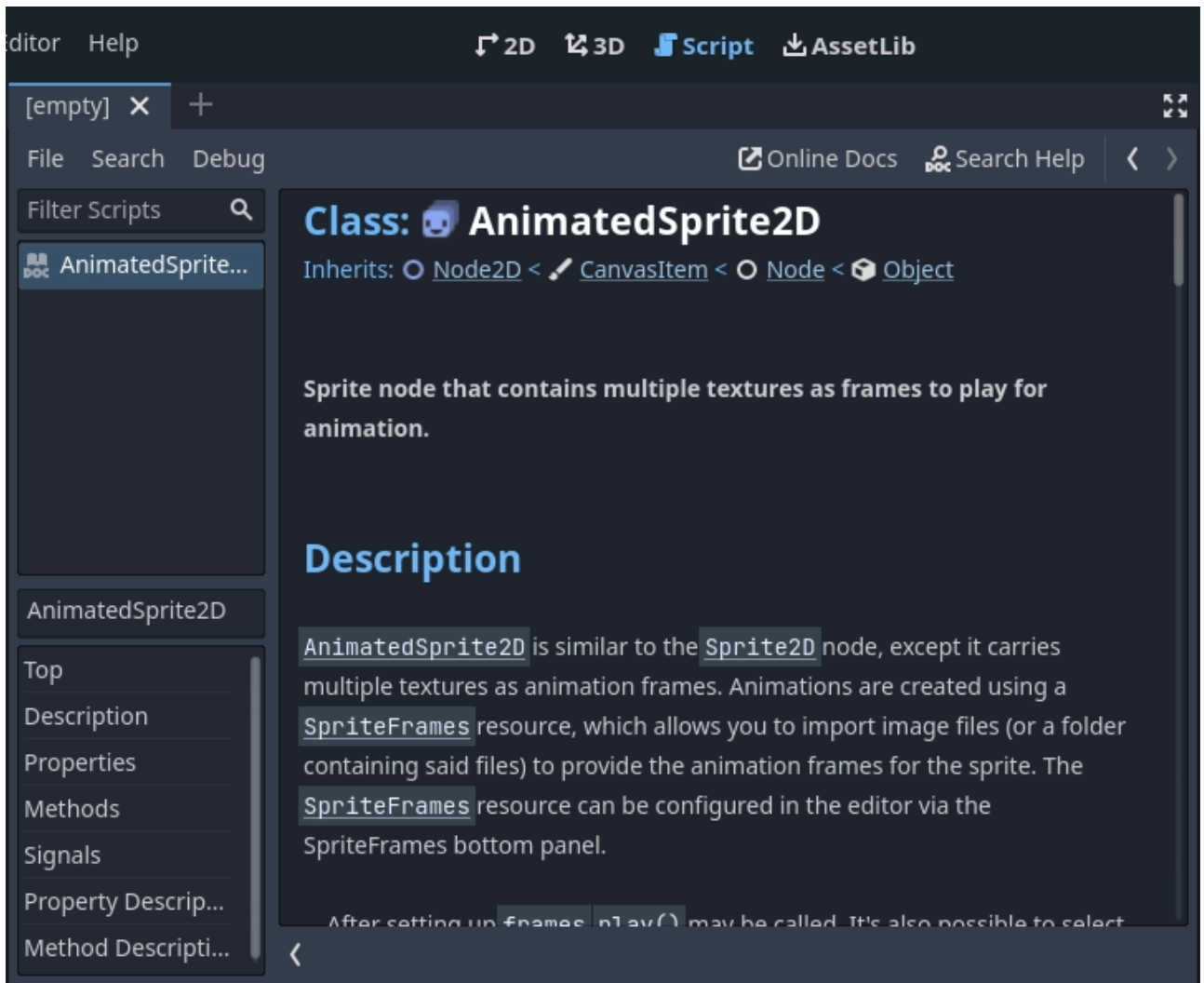
Вы можете найти информацию о классе, методе, свойстве, константе или сигнале одним из следующих способов:

- Нажатием клавиши F1 (или Opt + Space на macOS, или fn + F1 для ноутбуков с клавишей fn) в любом месте редактора.
 - Нажав кнопку "Search Help" ("Справка") в правом верхнем углу экрана написания скриптов.
 - Нажатием на "Help" в меню "Help".
 - Ctrl + Click по имени класса, функции, встроенной переменной в редакторе скриптов.
- Когда вы совершаете любое из этих действий, появляется окно. Введите текст для поиска любого элемента. Вы также можете использовать его для просмотра доступных объектов и методов.



Двойное нажатие по элементу откроет соответствующую страницу в экране

написания скриптов.



Философия дизайна Godot

Теперь, когда вы немного освоились, давайте поговорим о дизайне Godot.

Каждый игровой движок отличается и соответствует разным потребностям. Они не только предлагают широкий спектр функций, но и дизайн каждого движка уникален. Это приводит к разным рабочим процессам и различным способам формирования структур ваших игр. Все это вытекает из их соответствующих концепций дизайна.

Объектно-ориентированный дизайн и композиция

В основе Godot лежит объектно-ориентированный дизайн с его гибкой системой сцен и иерархией узлов. Он старается держаться подальше от строгих шаблонов программирования в угоду интуитивному способу структурирования вашей игры.

Например, Godot позволяет вам **составлять или собирать** сцены. Это как вложенные префабы: вы можете создать сцену с мерцающим светом и сцену со сломанным фонарём, который использует мерцающий свет. Затем создать город, наполненный сломанными фонарями. Поменяйте цвет мерцающего света, сохраните, и все сломанные фонари в городе тут же обновятся.

Более того, вы можете **наследовать** от любой сцены.

Сценой в Godot может являться оружие, персонаж, предмет, дверь, уровень, часть уровня... всё, что пожелаете. Это работает как классы в коде, за исключением лишь того, что вы создаёте всё сами, используя редактор, либо только код, либо и то и другое одновременно.

Это отличается от префабов, которые можно найти в некоторых 3D движках, так как вы можете наследовать и расширять любую сцену. К примеру, вы можете создать сцену "Маг", которая будет потомком вашей сцены "Персонаж". Измените сцену "Персонаж" в

редакторе, и сцена "Map" также будет обновлена. Это позволит вам построить свои проекты таким образом, чтобы их структура соответствовала дизайну игры.

Также обратите внимание, что Godot предлагает множество различных типов объектов, называемых узлами, каждый из которых имеет определенную цель. Узлы являются частью дерева и всегда наследуют от своих родителей вплоть до класса **Node (Узел)**. Хотя в движке и есть некоторые узлы, такие как формы столкновения, которые будут использоваться родительским физическим телом, большинство узлов работают независимо друг от друга.

Другими словами, узлы Godot работают не так, как компоненты в некоторых других игровых движках.

Sprite2D является Node2D, CanvasItem и Node. Он обладает всеми свойствами и особенностями трёх его родительских классов, такими как преобразования, или возможность рисовать собственные фигуры и рендер с использованием собственных шейдеров.